IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	Docket No.: **TI-32157**
Gilbert Cabillic	Art Unit: **2191**
Serial No.: **10/003,570**	Examiner: **Vo, Ted T.**
Filed: **10/24/2001**	Conf. No.: **5072**
For: **DATA PROCESSING APPARATUS, SYSTEM AND METHOD**	

APPELLANTS' BRIEF – 37 C.F.R. § 41.37

Commissioner for Patents

Alexandria, VA 22313-1450

Dear Sir:

This Appeal Brief is submitted in connection with the above-identified application in response to the final Office Action of March 18, 2010.

I.   REAL PARTY IN INTEREST

Texas Instruments Incorporated is the real party in interest.

II.   RELATED APPEALS AND INTERFERENCES

Appellants are not aware of an appeal in a related application.

III.   **STATUS OF CLAIMS**

Claims 16, 17 and 25-27 are canceled. Claims 1-15 and 18-24 are pending in the application. Rejection of Claims 1-15 and 18-24 was made by the Examiner in the Office Action dated March 18, 2010. <u>Claims 1-15 and 18-24 are on appeal.</u> Claims 1-15 and 18-24 are reproduced in the Appendix to Appellants' Brief filed herewith.

IV.   **STATUS OF AMENDMENTS**

An Amendment under 37 C.F.R. 1.116 was filed concurrently with this Appeal Brief. Appellants anticipate no issue with Examiner entering the amendment since the amendment places the case in better form for appeal by overcoming the objections to the specification and the 35 U.S.C. 101 rejection of Claims 1-10.

V.   **SUMMARY OF THE CLAIMED SUBJECT MATTER**

Claim 1 requires and positively recites, a method for generating in an apparatus program source code for translating high level code into instructions for a target processor, the method comprising:

determining a program code characteristic corresponding to said target processor (p. 14, line 33 – p. 15, line 5);

deriving one or more program code modules in accordance with said program code characteristic (p. 15, lines 5-9; p. 18, lines 11-18; p. 19, lines 30 – p. 20, line 6); and

generating program source code for translating high level code into instructions for said target processor from said one or more program code modules (p. 15, lines 5-11).

Claim 9 requires and positively recites, a method for creating in an apparatus program source code for translating between high level code and instructions for a target processor, comprising the steps of:

determining a program code characteristic corresponding to said target processor (p. 14, line 33 – p. 15, line 5);

selecting one or more predefined program code modules in accordance with said program code characteristic (p. 15, lines 5-9; p. 17, lines 12-14; p. 18, lines 11-18); and

forming program source code for translating high level code into instructions for said target processor from said selected one or more predefined program code modules (p. 15, lines 5-11).

Claim 11 requires and positively recites, a data processing apparatus for creating program source code for translating between high level code and instructions for a target processor P. 11, lines 21-27), the data processing apparatus being configured to:

determine a program code characteristic corresponding to said target processor identifier input to said data processing apparatus (p. 14, line 33 – p. 15, lie 5; p. 19, lines 19-28; p. 22, line 28 – p. 23, line 28);

derive one or more program code modules in accordance with said program code characteristic (p. 15, lines 5-9; p. 19, line 30 – p. 20, line 6); and

create program source code for translating high level code into instructions for said target processor from said derived one or more program code modules (p. 15, lines 5-11; p. 20, lines 8-12).

Claim 13 requires and positively recites, an apparatus (p. 11, lines 21-23), comprising at least one program source code module of a plurality of program source code modules for translating between high level code and instructions for a target processor (p. 15, lines 5-11; p. 22, lines 28-30), said at least one program code module corresponding to a characteristic of said target processor and being selected from said plurality of program source code modules (p. 15, lines 27-31; p. 18, lines 15-18; p. 26,

lines 16-25).

Claim 22 requires and positively recites, a processor (p. 11, lines 24-26), configured in accordance with program code comprising at least one program source code module of a plurality of program source code modules (p. 18, lines 11-14; p. 22, lines 12-15), for translating between high level code and instructions for a target processor (p. 15, lines 5-11; p. 22, lines 28-30), said at least one program source code module being in accordance with a characteristic of said target processor and selected from said plurality of program source code modules (p. 15, lines 27-31; p. 18, lines 15-18; p. 26, lines 16-25).

Claim 23 requires and positively recites, a processor (p. 22, lines 10-15), configured by program code comprising an agglomeration of two or more program source code modules of said plurality of said program source code modules (p. 22, lines 5-10).

Claim 24 requires and positively recites, a system (Fig. 1 (100) comprising a first (114a) and second (114c) processor (p. 11, lines 23-26), said first and second processor configured in accordance with program code comprising at least two program source code modules, wherein the first of said at least two program source code modules (p. 24, lines 14-15) is arranged to translate high level code to instructions for said first processor and a second of said at least two program source code modules (p. 24, lines 15-17) is arranged to translate high level code to instructions for said second processor (p. 15, lines 5-11; p. 16, lines 24-30; p. 24, lines 17-19; p. 25, lines 16-19).

VI.   GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1)   Are Claims 22-24 patentable under 35 U.S.C. 112, second paragraph?

2)   Are Claims 1-8, 9-10, 11-12, 13-15 and 18-21 patentable under 35 U.S.C.
101?

3)   Are Claims 1-15 and 18-24 patentable under 35 U.S.C. 102(b) over Smith et
al., "Distributed Programming with Intermediate IDL", June 1999, Ada Letters, Col. XIX,
No. 2, pages 90-95.

VII.   ARGUMENTS

1)   Claims 22-24 are rejected under 35 U.S.C. 112, second paragraph, as being
indefinite for failing to particularly point out and distinctly claim the subject matter which
applicant regards as the invention. Appellants traverse this rejection for the reasons set forth
below.

Appellants traverse Examiner's determination that, "None of the lements of a
processor is shown. It has been see in the claim that its recitation attempts using a method
for configuring the processor; its functionality is a process for use -- therefore it is unclear
whether the language of claim 22 and 23 is directed to a processor or a method of using that
processor" (OA, dated March 18, 2010, page 5, lines 8-11).

Appellants respectfully traverse Examiner's determination. Each of Claims 22 & 23
recite "A processor". Figure 1 of the specification clearly shows three processors --
processing unit 114a; processing unit 114c; and processing unit 114c. The specification
clearly sets forth, "System 100 may comprise a general purpose processor (114c), a digital
signal processor (DSP)(114a), and a hardware processor optimized for providing a virtual
machine platform for example" (page 11, lines 23-26). The specification further sets forth
processors 114a-114c on page 12, lines 7-8. The specification further sets forth that the
digital signal processor (DSP) can be a C55x DSP from Texas Instruments (page 22, lines

11-13, 28 -- page 23, line 28). The specification further identifies the general purpose processor as being an Intel Pentium II (page 22, lines 10-11; page 24, lines 17-18, 27-32; page 25, lines 21-22). Accordingly, the specification does identify the processors in the drawing and in the text sufficient to allow one of ordinary skill in the art to determine the elements of the respective processors.

Examiner is improperly requiring Appellants to limit their claims in a manner not required by law. Appellants are fully within their legal rights to draft apparatus claims with functional limitations. There is nothing indefinite about the scope of Claims 22 and 23 -- Examiner simply does not like the functional aspect of the claims. The legal citation set forth by Examiner is not applicable to this issue.

Moreover, Appellants respond with two Supreme Court cases in which the Court held that it is not necessary to recite in the claim everything necessary to operate the device. As stated by Joseph Gray Jackson in The Art of Drafting Patent Claims, 59-60:

> In Deering v. Winona, 155 U.S. 286 (1894), the device was an agricultural machine and lacked the support necessary for the board which was an element of the claim. The Supreme Court, said, "True that it is necessary and true it is not in the claim but it does not have to be; the claim does not have to include everything that is required to operate."

> The other case is Special Equipment v. Coe, 324 U.S. 370, 64 USPQ 525 (1945), in which a subcombination claim was supported which related to a machine for cutting, peeling and coring pears, and there was no cutting knife involved in the claim. The Supreme Court said it is perfectly all right; you do not have to have everything required to operate this device in the claim. Completeness is a much inflated "bugaboo" which is mainly of interest to certain examiners in the Patent Office, and should not really concern them. The claim is not a description of the device in any case. It is like a fingerprint which identifies the device. The fingerprint looks not at all like the person, but it is an identification of the person, and that is what we are interested in - identification.

If the Examiner is really rejecting Claims 22 & 23 because he feels the claims are overly broad, MPEP 706.03(d), paragraph 2, states, "The fact that a claim is broad does not necessarily justify a rejection on the ground that the claim is vague and indefinite or

incomplete. In non-chemical cases, a claim may, in general, be drawn as broadly as permitted by the prior art." Appellants submit that a broad claim, no matter how broad, is not indefinite as long as the boundaries of the claim are capable of being understood. The Examiner obviously understood the boundaries of Claims 22 & 23 since he rejected both under 35 U.S.C. 102(b) over Smith et al., "Distributed Programming with Intermediate IDL", June 1999, Ada Letters, Col. XIX, No. 2, pages 90-95. Accordingly, the 35 U.S.C. 112, second paragraph, rejection of Claims 22 & 23 is improper and must be reversed.

Appellants similarly traverse Examiner's determination that, "as per Claim 24, the claim recites a system comprising two processors. It shows that the claim merely recited a method for use, configured by upon two program source code modules. The claim is without any elements for structuring a system and/or the two of the processors. Therefore, it is unclear whether the language of claim 24 is directed to a system of two processors or a method for use" (OA, dated March 18, 2010, page 5, line 18 – page 6, line 2). Examiner is improperly requiring Appellants to limit their claims in a manner not required by law. Appellants are fully within their legal rights to draft apparatus claims with functional limitations. There is nothing indefinite about the scope of Claim 24 – Examiner simply does not like the functional aspect of the claim. The legal citation set forth by Examiner is not applicable to this issue.

As set forth above, Figure 1 of the specification clearly shows three processors – processing unit 114a; processing unit 114c; and processing unit 114c. The specification clearly sets forth, "System 100 may comprise a general purpose processor (114c), a digital signal processor (DSP)(114a), and a hardware processor optimized for providing a virtual machine platform for example" (page 11, lines 23-26). The specification further sets forth processors 114a-114c on page 12, lines 7-8. The specification further sets forth that the digital signal processor (DSP) can be a C55x DSP from Texas Instruments (page 22, lines 11-13, 28 – page 23, line 28). The specification further identifies the general purpose

processor as being an Intel Pentium II (page 22, lines 10-11; page 24, lines 17-18, 27-32; page 25, lines 21-22). Thus, the "first processor" of Claim 24 can be DSP 114(a) and the "second processor" of Claim 24 can be general purpose processor 114(c) or visa versa. Accordingly, the specification does identify the processors in the drawing and in the text sufficient to allow one of ordinary skill in the art to determine the elements of the respective processors. Thus, Claim 24 does show elements for structuring a system and/or the two of the processors.

Appellants similarly point to two Supreme Court cases in which the Court held that it is not necessary to recite in the claim everything necessary to operate the device. As stated by Joseph Gray Jackson in The Art of Drafting Patent Claims, 59-60:

> In Deering v. Winona, 155 U.S. 286 (1894), the device was an agricultural machine and lacked the support necessary for the board which was an element of the claim. The Supreme Court, said, "True that it is necessary and true it is not in the claim but it does not have to be; the claim does not have to include everything that is required to operate."
>
> The other case is Special Equipment v. Coe, 324 U.S. 370, 64 USPQ 525 (1945), in which a subcombination claim was supported which related to a machine for cutting, peeling and coring pears, and there was no cutting knife involved in the claim. The Supreme Court said it is perfectly all right; you do not have to have everything required to operate this device in the claim. Completeness is a much inflated "bugaboo" which is mainly of interest to certain examiners in the Patent Office, and should not really concern them. The claim is not a description of the device in any case. It is like a fingerprint which identifies the device. The fingerprint looks not at all like the person, but it is an identification of the person, and that is what we are interested in - identification.

If the Examiner is really rejecting Claim 24 because he feels the claims are overly broad, MPEP 706.03(d), paragraph 2, states, "The fact that a claim is broad does not necessarily justify a rejection on the ground that the claim is vague and indefinite or incomplete. In non-chemical cases, a claim may, in general, be drawn as broadly as permitted by the prior art." Appellants submit that a broad claim, no matter how broad, is not indefinite as long as the boundaries of the claim are capable of being understood. The

Examiner obviously understood the boundaries of Claim 24 since he rejected it under 35 U.S.C. 102(b) over Smith et al., "Distributed Programming with Intermediate IDL", June 1999, Ada Letters, Col. XIX, No. 2, pages 90-95. Accordingly, the 35 U.S.C. 112, second paragraph, rejection of Claim 24 is improper and must be reversed.

2)   Claims 1-8, 9-10, 11-12, 13-15 and 18-21 stand rejected under 35 U.S.C. § 101 "because the claimed invention is directed to non-statutory subject matter". Appellants traverse this rejection for the reasons set forth below.

Based on U.S. Supreme Court precedent (*Diamond v. Diehr*, 450 U.S. 175, 184 (1981); *Parker v. Flook*, 437 U.S. 584, 588 n.9 (1978); *Gottschalk v. Benson*, 409 U.S. 63, 70 (1972); *Cochrane v. Deener*, 94 U.S. 780, 787-88 (1876) and recent Federal Circuit decisions, a § 101 process must (1) be tied to another statutory class (such as a particular apparatus) **OR** (2) **transform underlying subject matter (such as an article or materials) to a different state or thing** (The Supreme Court recognized that his test is not necessarily fixed or permanent and may evolve with technological advances. *Gottschalk v. Benson*, 409 U.S. 63, 71 (1972).

Independent Claim 1, as amended, requires and positively recites, a method for **generating _in an apparatus_ program source code** for translating high level code into instructions for a target processor, the method comprising: "determining a **program code characteristic corresponding to said target processor**", "deriving one or more program code **modules** in accordance with said program code characteristic" and "**generating program source code** for translating high level code into instructions for said target processor from said one or more program code modules".

The preamble of Claim 1 recites, "a method for **generating _in an apparatus_ program source code** for translating high level code into instructions for a target

processor". Thus, Claim 1 has been amended to definitively tie it to a statutory subject matter. This is clearly a tie-in of the method to an apparatus which is a recognized statutory class. The above language clearly ties the method to another statutory class (such as a particular apparatus) which qualifies under Supreme Court and Federal Circuit precedent "(1) be tied to another statutory class (such as a particular apparatus) above as being an eligible process under 35 U.S.C. § 101. For this reason alone the 35 U.S.C. § 101 rejection of Claim 1 is improper and must be reversed.

Moreover, Claim 1, as amended, now prohibits handwriting from forming the method since the method must be formed in an apparatus.

In the event Examiner determines not to give any patentable weight to the preamble, Appellants respond as follows:

The determination of whether a preamble limits a claim is made on a case-by-case basis in light of the facts in each case; there is no litmus test defining when a preamble limits the scope of a claim. _Catalina Mktg. Int'l v. Coolsavings.com, Inc._, 289 F.3d 801, 808, 62 USPQ2d 1781, 1785 (Fed. Cir. 2002). See _id._ at 808-10, 62 USPQ2d at 1784-86 for a discussion of guideposts that have emerged from various decisions exploring the preamble's effect on claim scope, as well as a hypothetical example illustrating these principles.

"[A] Claim preamble has the import that the claim as a whole suggests for it." _Bell Communications Research, Inc. v. Vitalink Communications Corp._, 55 F.3d 615, 620, 34 USPQ2d 1816, 1820 (Fed. Cir. 1995). "If the claim preamble, when read in the context of the entire claim, **recites limitations of the claim**, or, if claim preamble is **'necessary to give life, meaning, and vitality' to the claim, then the claim preamble should be construed as if in the balance of the claim.**" _Pitney Bowes, Inc. v. Hewlett-Packard Co._, 182 F.3d 1298, 1305, 51 USPQ2d 1161, 1165-66 (Fed. Cir. 1999). See also

Jansen v. Rexall Sundown, Inc., 342 F.3d 1329, 1333, 68 USPQ2d 1154, 1158 (Fed. Cir. 2003)(In considering the effect of the preamble in a claim directed to a method of treating or preventing pernicious anemia in humans by administering a certain vitamin preparation to "a human in need thereof," the court held that the claims' recitation of a patient or a human "in need" gives life and meaning to the preamble's statement of purpose). *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951) (A preamble reciting "An abrasive article" was deemed essential to point out the invention defined by claims to an article comprising abrasive grains and a hardened binder and the process of making it. The court stated "it is only by that phrase that it can be known that the subject matter defined by the claims is comprised as an abrasive article. Every union of substances capable *inter alia* of use as abrasive grains and a binder is not an 'abrasive article.'" Therefore, the preamble served to further define the structure of the article produced.).

Any terminology in the preamble that limits the structure of the claimed invention must be treated as a claim limitation. See, e.g., *Corning Glass Works v. Sumitomo Elec. U.S.A., Inc.*, 868 F.2d 1251, 1257, 9 USPQ2d 1962, 1966 (Fed. Cir. 1989) (The determination of whether preamble recitations are structural limitations can be resolved **only on review of the entirety of the application "to gain an understanding of what the inventors actually invented and intended to encompass by the claim.");** *Pac-Tec Inc. v. Amerace Corp.*, 903 F.2d 796, 801, 14 USPQ2d 1871, 1876 (Fed. Cir. 1990) (determining that preamble language that constitutes a structural limitation is actually part of the claimed invention). See also *In re Stencel*, 828 F.2d 751, 4 USPQ2d 1071 (Fed. Cir. 1987). (The claim at issue was directed to a driver for setting a joint of a threaded collar*>;< however>,< the body of the claim did not directly include the structure of the collar as part of the claimed article. The examiner did not consider the preamble, which did set forth the structure of the collar, as limiting the claim. The court found that the collar structure could not be ignored. While the claim was not directly limited to the collar, the collar structure recited in the preamble did limit the

structure of the driver. "[T]he framework - the teachings of the prior art - against which patentability is measured is not all drivers broadly, but drivers suitable for use in combination with this collar, for the claims are so limited." Id. at 1073, 828 F.2d at 754.).

Being that the novelty of the present invention is directed to a method for **generating _in an apparatus_ program source code for translating high level code into instructions for a target processor**, the preamble limits the structure of the claimed invention and must be treated as a claim limitation. See, e.g., _Corning Glass Works v. Sumitomo Elec. U.S.A., Inc._, 868 F.2d 1251, 1257, 9 USPQ2d 1962, 1966 (Fed. Cir. 1989).

In addition to the above, the limitation in Claim 1, "**generating _in an apparatus_ program source code for translating high level code into instructions for said target processor from said one or more program code modules**", clearly recites a transformation which qualifies under Supreme Court precedent "**OR** (2) **transform underlying subject matter (such as an article or materials) to a different state or thing**" above as being an eligible process under 35 U.S.C. § 101. For this reason alone, or in combination with the above argument, the 35 U.S.C. § 101 rejection of Claim 1 is improper and must be reversed.

Independent Claim 9, as amended, requires and positively recites, a method for **creating _in an apparatus_ program source code for translating between high level code and instructions for a target processor**, comprising the steps of: "**determining a program code characteristic corresponding to said target processor**", "**selecting one or more predefined program code modules in accordance with said program code characteristic**" and "**forming program source code for translating high level code into instructions for said target processor from said selected one or more predefined program code modules**".

The preamble of Claim 9 recites, "a method for **creating** *in an apparatus* **program source code** for **translating high level code into instructions for a target processor**". Thus, Claim 9 has been amended to definitively tie it to a statutory subject matter. This is clearly a tie-in of the method to an apparatus which is a recognized statutory class. The above language clearly ties the method to another statutory class (such as a particular apparatus) which qualifies under Supreme Court and Federal Circuit precedent "(1) be tied to another statutory class (such as a particular apparatus) above as being an eligible process under 35 U.S.C. § 101. For this reason alone the 35 U.S.C. § 101 rejection of Claim 9 is improper and must be reversed.

Moreover, Claim 9, as amended, now prohibits handwriting from forming the method since the method must be formed in an apparatus.

In the event Examiner determines not to give any patentable weight to the preamble, Appellants respond as follows:

The determination of whether a preamble limits a claim is made on a case-by-case basis in light of the facts in each case; there is no litmus test defining when a preamble limits the scope of a claim. *Catalina Mktg. Int'l v. Coolsavings.com, Inc.*, 289 F.3d 801, 808, 62 USPQ2d 1781, 1785 (Fed. Cir. 2002). See *id.* at 808-10, 62 USPQ2d at 1784-86 for a discussion of guideposts that have emerged from various decisions exploring the preamble's effect on claim scope, as well as a hypothetical example illustrating these principles.

"[A] Claim preamble has the import that the claim as a whole suggests for it." *Bell Communications Research, Inc. v. Vitalink Communications Corp.*, 55 F.3d 615, 620, 34 USPQ2d 1816, 1820 (Fed. Cir. 1995). "If the claim preamble, when read in the context of the entire claim, **recites limitations of the claim**, or, if the claim preamble is **'necessary to give life, meaning, and vitality'** to the claim, then the claim preamble

should be construed as if in the balance of the claim." *Pitney Bowes, Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298, 1305, 51 USPQ2d 1161, 1165-66 (Fed. Cir. 1999). See also *Jansen v. Rexall Sundown, Inc.*, 342 F.3d 1329, 1333, 68 USPQ2d 1154, 1158 (Fed. Cir. 2003)(In considering the effect of the preamble in a claim directed to a method of treating or preventing pernicious anemia in humans by administering a certain vitamin preparation to "a human in need thereof," the court held that the claims' recitation of a patient or a human "in need" gives life and meaning to the preamble's statement of purpose.); *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951) (A preamble reciting "An abrasive article" was deemed essential to point out the invention defined by claims to an article comprising abrasive grains and a hardened binder and the process of making it. The court stated "it is only by that phrase that it can be known that the subject matter defined by the claims is comprised as an abrasive article. Every union of substances capable *inter alia* of use as abrasive grains and a binder is not an 'abrasive article.'" Therefore, the preamble served to further define the structure of the article produced.).

Any terminology in the preamble that limits the structure of the claimed invention must be treated as a claim limitation. See, e.g., *Corning Glass Works v. Sumitomo Elec. U.S.A., Inc.*, 868 F.2d 1251, 1257, 9 USPQ2d 1962, 1966 (Fed. Cir. 1989) (The determination of whether preamble recitations are structural limitations can be resolved only on **review of the entirety of the application "to gain an understanding of what the inventors actually invented and intended to encompass by the claim.**"); *Pac-Tec Inc. v. Amerace Corp.*, 903 F.2d 796, 801, 14 USPQ2d 1871, 1876 (Fed. Cir. 1990) (determining that preamble language that constitutes a structural limitation is actually part of the claimed invention). See also *In re Stencel*, 828 F.2d 751, 4 USPQ2d 1071 (Fed. Cir. 1987). (The claim at issue was directed to a driver for setting a joint of a threaded collar*>;< however>,< the body of the claim did not directly include the structure of the collar as part of the claimed article. The examiner did not consider the preamble, which did set forth the structure of the collar, as limiting the claim. The

court found that the collar structure could not be ignored. While the claim was not directly limited to the collar, the collar structure recited in the preamble did limit the structure of the driver. "[T]he framework - the teachings of the prior art - against which patentability is measured is not all drivers broadly, but drivers suitable for use in combination with this collar, for the claims are so limited." Id. at 1073, 828 F.2d at 754.).

Being that the novelty of the present invention is directed to a method for **creating** program source code **for translating high level code into instructions for a target processor**, the preamble limits the structure of the claimed invention and must be treated as a claim limitation. See, e.g., *Corning Glass Works v. Sumitomo Elec. U.S.A., Inc.,* 868 F.2d 1251, 1257, 9 USPQ2d 1962, 1966 (Fed. Cir. 1989).

In addition to the above, the limitation in Claim 9, "**forming program source code for translating high level code into instructions for said target processor from said selected one or more predefined program code modules**", clearly recites a transformation which qualifies under Supreme Court precedent "**OR** (2) **transform underlying subject matter (such as an article or materials) to a different state or thing**" above as being an eligible process under 35 U.S.C. § 101. For this reason alone, or in combination with the above argument, the 35 U.S.C. § 101 rejection of Claim 9 is improper and must be reversed.

Independent Claim 11, as amended, requires and positively recites, a **data processing apparatus** for **creating** program source code **for translating between high level code and instructions for a target processor**, the **data processing apparatus being configured to**: "determine a **program code characteristic corresponding to said target processor identifier** input to said data processing apparatus", "derive one or more program code **modules** **in accordance with said program code characteristic**" and "**create program source code for translating high level code into instructions for said**

target processor from said derived one or more program code modules".

The preamble of Claim 11 recites, "a **data processing apparatus** for **creating program source code** for translating between high level code and instructions for a target processor, the **data processing apparatus being configured to**". An "apparatus" is clearly a recognized statutory class.

Moreover, the language ""a **data processing apparatus** for **creating** program source code …" clearly requires that the program source code be created in the data processing apparatus. The above language clearly qualifies under Supreme Court and Federal Circuit precedent "(1) be tied to another statutory class (such as a particular apparatus) above as being an eligible process under 35 U.S.C. § 101. Moreover, Examiner must consider ALL of the words of Claim 11 -- not just the words Examiner wants to consider. For this reason alone the 35 U.S.C. § 101 rejection of Claim 11 is improper and must be reversed.

In the event Examiner determines not to give any patentable weight to the preamble, Appellants respond as follows:

The determination of whether a preamble limits a claim is made on a case-by-case basis in light of the facts in each case; there is no litmus test defining when a preamble limits the scope of a claim. *Catalina Mktg. Int'l v. Coolsavings.com, Inc.*, 289 F.3d 801, 808, 62 USPQ2d 1781, 1785 (Fed. Cir. 2002). See *id.* at 808-10, 62 USPQ2d at 1784-86 for a discussion of guideposts that have emerged from various decisions exploring the preamble's effect on claim scope, as well as a hypothetical example illustrating these principles.

"[A] Claim preamble has the import that the claim as a whole suggests for it." *Bell Communications Research, Inc. v. Vitalink Communications Corp.*, 55 F.3d 615, 620, 34

USPQ2d 1816, 1820 (Fed. Cir. 1995). "If the claim preamble, when read in the context of the entire claim, **recites limitations of the claim**, or, if the claim preamble is **'necessary to give life, meaning, and vitality' to the claim, then the claim preamble should be construed as if in the balance of the claim.**" *Pitney Bowes, Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298, 1305, 51 USPQ2d 1161, 1165-66 (Fed. Cir. 1999). See also *Jansen v. Rexall Sundown, Inc.*, 342 F.3d 1329, 1333, 68 USPQ2d 1154, 1158 (Fed. Cir. 2003)(In considering the effect of the preamble in a claim directed to a method of treating or preventing pernicious anemia in humans by administering a certain vitamin preparation to "a human in need thereof," the court held that the claims' recitation of a patient or a human "in need" gives life and meaning to the preamble's statement of purpose). *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951) (A preamble reciting "An abrasive article" was deemed essential to point out the invention defined by claims to an article comprising abrasive grains and a hardened binder and the process of making it. The court stated "it is only by that phrase that it can be known that the subject matter defined by the claims is comprised as an abrasive article. Every union of substances capable *inter alia* of use as abrasive grains and a binder is not an 'abrasive article.'" Therefore, the preamble served to further define the structure of the article produced.).

Any terminology in the preamble that limits the structure of the claimed **invention must be treated as a claim limitation.** See, e.g., *Corning Glass Works v. Sumitomo Elec. U.S.A., Inc.*, 868 F.2d 1251, 1257, 9 USPQ2d 1962, 1966 (Fed. Cir. 1989) (The determination of whether preamble recitations are structural limitations can be resolved **only on review of the entirety of the application "to gain an understanding of what the inventors actually invented and intended to encompass by the claim.");** *Pac-Tec Inc. v. Amerace Corp.*, 903 F.2d 796, 801, 14 USPQ2d 1871, 1876 (Fed. Cir. 1990) (determining that preamble language that constitutes a structural limitation is actually part of the claimed invention). See also *In re Stencel*, 828 F.2d 751, 4 USPQ2d 1071 (Fed. Cir. 1987). (The claim at issue was directed to a driver for setting a

joint of a threaded collar*>;< however>,< the body of the claim did not directly include the structure of the collar as part of the claimed article. The examiner did not consider the preamble, which did set forth the structure of the collar, as limiting the claim. The court found that the collar structure could not be ignored. While the claim was not directly limited to the collar, the collar structure recited in the preamble did limit the structure of the driver. "[T]he framework - the teachings of the prior art - against which patentability is measured is not all drivers broadly, but drivers suitable for use in combination with this collar, for the claims are so limited." Id. at 1073, 828 F.2d at 754.).

Being that the novelty of the present invention is directed to a **data processing apparatus** for **creating** program source code **for translating between high level code and instructions for a target processor**, the **data processing apparatus being configured to ...**, the preamble limits the structure of the claimed invention and must be treated as a claim limitation. See, e.g., *Corning Glass Works v. Sumitomo Elec. U.S.A., Inc.,* 868 F.2d 1251, 1257, 9 USPQ2d 1962, 1966 (Fed. Cir. 1989).

In addition to the above, the limitation in Claim 11, "**create program source code for translating high level code into instructions for said target processor from said derived one or more program code modules**", clearly recites a transformation which qualifies under Supreme Court precedent "**OR** (2) **transform underlying subject matter (such as an article or materials) to a different state or thing**" above as being an eligible process under 35 U.S.C. § 101. For this reason alone, or in combination with the above argument, the 35 U.S.C. § 101 rejection of Claim 11 is improper and must be reversed.

Independent Claim 13, as amended, requires and positively recites, **an apparatus**, comprising **at least one program source code module** of **a plurality of program source code modules** for **translating between high level code and instructions for a**

target processor, said at least one **program source code module corresponding to a characteristic of said target processor** and **being selected from** said plurality of program code modules.

The preamble of Claim 13 recites, "an apparatus". An "apparatus" is clearly a recognized statutory class. The above language clearly qualifies under Supreme Court and Federal Circuit precedent "(1) be tied to another statutory class (such as a particular apparatus) above as being an eligible process under 35 U.S.C. § 101. For this reason alone the 35 U.S.C. § 101 rejection of Claim 13 is improper and must be reversed.

Moreover, Claim 13 further recites, "**at least one program source code module ... for translating between high level code and instructions for a target processor**". Thus, even if, arguendo, if the term apparatus were insufficient, the above language requires that it is the source code module that translates between high level code and instructions for a target processor – which prohibits the Examiner's "hand written" translation argument.

In addition to the above, the additional language in Claim 13, the language "**program source code module corresponding to a characteristic of said target processor**" signals that the module reacts to a characteristic of the target processor, which qualifies as a transformation.

In the event Examiner determines not to give any patentable weight to the preamble, Appellants respond as follows:

The determination of whether a preamble limits a claim is made on a case-by-case basis in light of the facts in each case; there is no litmus test defining when a preamble limits the scope of a claim. *Catalina Mktg. Int'l v. Coolsavings.com, Inc.*, 289 F.3d 801, 808, 62 USPQ2d 1781, 1785 (Fed. Cir. 2002). See *id.* at 808-10, 62 USPQ2d at 1784-86

for a discussion of guideposts that have emerged from various decisions exploring the preamble's effect on claim scope, as well as a hypothetical example illustrating these principles.

"[A] Claim preamble has the import that the claim as a whole suggests for it." *Bell Communications Research, Inc. v. Vitalink Communications Corp.*, 55 F.3d 615, 620, 34 USPQ2d 1816, 1820 (Fed. Cir. 1995). "If the claim preamble, when read in the context of the entire claim, **recites limitations of the claim**, or, if the claim preamble is **'necessary to give life, meaning, and vitality' to the claim, then the claim preamble should be construed as if in the balance of the claim.**" *Pitney Bowes, Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298, 1305, 51 USPQ2d 1161, 1165-66 (Fed. Cir. 1999). See also *Jansen v. Rexall Sundown, Inc.*, 342 F.3d 1329, 1333, 68 USPQ2d 1154, 1158 (Fed. Cir. 2003)(In considering the effect of the preamble in a claim directed to a method of treating or preventing pernicious anemia in humans by administering a certain vitamin preparation to "a human in need thereof," the court held that the claims' recitation of a patient or a human "in need" gives life and meaning to the preamble's statement of purpose). *Kropa v. Robie*, 187 F.2d 150, 152, 88 USPQ 478, 481 (CCPA 1951) (A preamble reciting "An abrasive article" was deemed essential to point out the invention defined by claims to an article comprising abrasive grains and a hardened binder and the process of making it. The court stated "it is only by that phrase that it can be known that the subject matter defined by the claims is comprised as an abrasive article. Every union of substances capable *inter alia* of use as abrasive grains and a binder is not an 'abrasive article.'" Therefore, the preamble served to further define the structure of the article produced.).

Any terminology in the preamble that limits the structure of the claimed invention must be treated as a claim limitation. See, e.g., *Corning Glass Works v. Sumitomo Elec. U.S.A., Inc.*, 868 F.2d 1251, 1257, 9 USPQ2d 1962, 1966 (Fed. Cir. 1989) (The determination of whether preamble recitations are structural limitations can

be resolved only **on review of the entirety of the application "to gain an understanding of what the inventors actually invented and intended to encompass by the claim.");** _Pac-Tec Inc. v. Amerace Corp._, 903 F.2d 796, 801, 14 USPQ2d 1871, 1876 (Fed. Cir. 1990) (determining that preamble language that constitutes a structural limitation is actually part of the claimed invention). See also _In re Stencel_, 828 F.2d 751, 4 USPQ2d 1071 (Fed. Cir. 1987). (The claim at issue was directed to a driver for setting a joint of a threaded collar*>;< however>,< the body of the claim did not directly include the structure of the collar as part of the claimed article. The examiner did not consider the preamble, which did set forth the structure of the collar, as limiting the claim. The court found that the collar structure could not be ignored. While the claim was not directly limited to the collar, the collar structure recited in the preamble did limit the structure of the driver. "[T]he framework - the teachings of the prior art - against which patentability is measured is not all drivers broadly, but drivers suitable for use in combination with this collar, for the claims are so limited." Id. at 1073, 828 F.2d at 754.).

Being that the novelty of the present invention is directed to an apparatus, the preamble limits the structure of the claimed invention and must be treated as a claim limitation. See, e.g., _Corning Glass Works v. Sumitomo Elec. U.S.A., Inc._, 868 F.2d 1251, 1257, 9 USPQ2d 1962, 1966 (Fed. Cir. 1989).

In addition to the above, the limitation in Claim 13, "**at least one program source code module of a plurality of program source code modules** for **translating between high level code and instructions for a target processor**, said at least one program source code **module corresponding to a characteristic of said target processor** and **being selected from said plurality of program code modules**", clearly recites a transformation which qualifies under Supreme Court precedent "**OR** (2) **transform underlying subject matter (such as an article or materials) to a different state or thing**" above as being an eligible process under 35 U.S.C. § 101. For this reason

alone, or in combination with the above argument, the 35 U.S.C. § 101 rejection of Claim 13 is improper and must be reversed.

Claims 2-8 depend directly, or indirectly, from independent Claim 1. Claim 10 depends directly from Claim 9. Claim 12 depends directly from Claim 11. Claims 14, 15 and 18-21 depend directly, or indirectly, from Claim 13. The same arguments set forth above similarly apply to these claims.

3)   Claims 1-15 and 18-24 stand rejected under 35 U.S.C. 102(b) as being anticipated by Smith et al., "Distributed Programming with Intermediate IDL", June 1999, Ada Letters, Col. XIX, No. 2, pages 90-95. Appellants respectfully traverse this rejection as set forth below.

In order that the rejection of Claims 1-15 and 18-24 be sustainable, it is fundamental that "each and every element as set forth in the claim be found, either expressly or inherently described, in a single prior art reference." _Verdegall Bros. v. Union Oil Co. of California_, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). See also, _Richardson v. Suzuki Motor Co._, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989), where the court states, "The identical invention must be shown in as complete detail as is contained in the ... claim".

Furthermore, "all words in a claim must be considered in judging the patentability of that claim against the prior art." _In re Wilson_, 424 F.2d 1382, 1385, 165 USPQ 494, 496 (CCPA 1970).

Independent Claim 1, as amended, requires and positively recites, a method for **generating in an apparatus program source code for translating high level code into instructions for a target processor**, the method comprising: "**determining a program code characteristic corresponding to said target processor**", "deriving one or more

program code modules in accordance with said program code characteristic" and "generating program source code **for translating high level code <u>into instructions for said target processor from said one or more program code modules</u>**".

Independent Claim 9, as amended, requires and positively recites, a method for **creating in an apparatus program source code** for **translating between high level code and <u>instructions for a target processor</u>**, comprising the steps of: "**determining a program code characteristic <u>corresponding to said target processor</u>**", "selecting one or more predefined program code modules in accordance with said program code characteristic" and "forming program source code **for translating high level code <u>into instructions for said target processor from said selected one or more predefined program code modules</u>**".

Independent Claim 11, as amended, requires and positively recites, <u>a data processing apparatus</u> for creating program source code for **translating between high level code and <u>instructions for a target processor</u>**, the <u>data processing apparatus being configured to</u>: "determine a program code characteristic <u>corresponding to said target processor identifier input to said data processing apparatus</u>", "derive one or more program code modules in accordance with said program code characteristic" and "**create program source code for translating high level code <u>into instructions for said target processor from said derived one or more program code modules</u>**".

Independent Claim 13, as amended, requires and positively recites, an <u>apparatus</u>, comprising at least one program source code module of a plurality of program source code modules for **translating between high level code and instructions <u>for a target processor</u>**, said at least one program source code module <u>corresponding to a characteristic of said target processor</u> and being selected from said plurality of program code modules.

Independent Claim 22, as amended, requires and positively recites, a **processor**, configured in accordance with program code comprising at least one program source code module of a plurality of program source code modules, for **translating between high level code and instructions for a target processor**, said at least one program source code module **being in accordance with a characteristic of said target processor** and selected from said plurality of program source code modules.

Independent Claim 23, as amended, requires and positively recites, a **processor**, configured by **program code** comprising **an agglomeration of two or more program source code modules** of said plurality of said program code modules.

Independent Claim 24, as amended, requires and positively recites, a **system** comprising **a first and second processor**, **said first and second processor configured in accordance with** program code comprising at least two program source code modules, wherein the **first of said at least two program source code modules is arranged to translate high level code to instructions for said first processor** and **a second of said at least two program source code modules is arranged to translate high level code to instructions for said second processor**.

In contrast, Smith teaches an approach to heterogeneous-language distributed programming where an implicit hidden IDL specification is used as an intermediate step in the translation process from server language specification to client-language specification (p. 95, col. 1, lines 29-34). Smith teaches that, "source translations convert the specification of a server object into a corresponding representation in an Intermediate Object Specification (IOS) language" and "a target translation converts the IOS representation into a client's language" (p. 92, lines 7-11). Yet no where is there any teaching in Smith that its "heterogeneous-language distributed programming" is applicable to any processor. As such, Smith fails to teach or suggest, a method for **generating in an apparatus program source code** for **translating high level code into instructions for a target processor**,

the method comprising: "**determining a program code characteristic <u>corresponding to</u> <u>said target processor</u>**", "deriving one or more program code modules in accordance with said program code characteristic" and "generating program source code **for translating high level code <u>into instructions for said target processor from said one or more</u> <u>program code modules</u>**", as required by Claim 1, OR a method for **creating in an apparatus program source code** for **translating between high level code and instructions for a target processor**, comprising the steps of: "**determining a program code characteristic <u>corresponding to said target processor</u>**", "selecting one or more predefined program code modules in accordance with said program code characteristic" and "forming program source code **for translating high level code <u>into instructions for</u> <u>said target processor from said selected one or more predefined program code</u> <u>modules</u>**", as required by Claim 9.

Smith similarly fails to teach or suggest, **<u>a data processing apparatus</u>** for creating program source code for **translating between high level code and <u>instructions for a</u> <u>target processor</u>**, the **<u>data processing apparatus being configured to</u>**: "determine a program code characteristic **<u>corresponding to said target processor identifier input to</u> <u>said data processing apparatus</u>**", "derive one or more program code modules in accordance with said program code characteristic" and "**create program source code for translating high level code <u>into instructions for said target processor from said</u> <u>derived one or more program code modules</u>**", as required by Claim 11, OR "an **apparatus**, comprising at least one program source code module of a plurality of program source code modules for **translating between high level code and instructions <u>for a target processor</u>**, said at least one program source code module **<u>corresponding to</u> <u>a characteristic of said target processor</u>** and being selected from said plurality of program code modules", as required by Claim 13, OR "a **processor**, configured in accordance with program code comprising at least one program source code module of a plurality of program source code modules, for **translating between high level code and instructions <u>for a target processor</u>**, said at least one program source code module **<u>being</u>**

in accordance with a characteristic of said target processor and selected from said plurality of program source code modules", as required by Claim 22, OR "a **processor**, configured by **program code** comprising **an agglomeration of two or more program source code modules** of said plurality of said program code modules", as required by Claim 23, OR "a **system** comprising **a first and second processor**, said first and second **processor configured in accordance with** program code comprising at least two program source code modules, wherein the **first of said at least two program source code modules is arranged to translate high level code to instructions _for said first processor_** and **a second of said at least two program source code modules is arranged to translate high level code to instructions _for said second processor_**", as required by Claim 24. Accordingly, the 35 U.S.C. 102(b) rejection of Claims 1, 9, 11, 13, 22, 23 and 24 is improper and must be reversed.

Claims 2-8 stand allowable as depending (directly or indirectly) from allowable Claim 1 and including further limitations not taught or suggested by the references of record. Claim 10 stands allowable as depending from allowable Claim 9 and including further limitations not taught or suggested by the references of record. Claim 12 stands allowable as depending from allowable Claim 11 and including further limitations not taught or suggested by the references of record. Claims 14, 15 and 18-21 stand allowable as depending (directly or indirectly) from allowable Claim 13 and including further limitations not taught or suggested by the references of record.

Claim 2 further defines the method according to claim 1, by generating program source code for **translating high level code into instructions _for one of a plurality of target processors_**. Claim 2 is allowable for the same reasons provided above in support of the allowability of Claim 1. In addition to the above, there is no teaching whatsoever in Smith for a "plurality" of target processors. As such, Smith fails to teach or suggest, "generating program source code for **translating high level code into instructions _for one of a plurality of target processors_**", as required by Claim 2. As such, Claim 2 is

not anticipated by Smith. Accordingly, the 35 U.S.C. 102(b) rejection is improper and must be reversed.

Claim 3 further defines the method according to claim 1, by further comprising forming agglomerated program source code from a plurality of program code modules in accordance with said program code characteristic. Claim 3 is allowable for the same reasons provided above in support of the allowability of Claim 1. Accordingly, the 35 U.S.C. 102(b) rejection is improper and must be reversed.

Claim 4 further defines the method according to claim 1, by further comprising deriving said program code modules **in accordance with <u>a desired functionality for said target processor</u>**. Claim 4 is allowable for the same reasons provided above in support of the allowability of Claim 1. Claim 4 is allowable for the same reasons provided above in support of the allowability of Claim 1. In addition to the above, Smith discloses translating "server language specification to corresponding client language representations". There is no mention of "a processor" in Smith, much less the additional requirement of "deriving said program code modules **in accordance with <u>a desired functionality for said target processor</u>**". As such, Claim 4 is not anticipated by Smith. Accordingly, the 35 U.S.C. 102(b) rejection is improper and must be reversed.

Claim 5 further defines the method according to claim 2, wherein: "said step of determining comprises **determining respective program code characteristics <u>for respective ones of a plurality of target processors</u>**", "said step of deriving comprises deriving respective program code modules in accordance with said respective program code characteristics" and "said step of generating comprises generating program source code for translating high level code into instructions **<u>for said target processors</u>** from said program code modules". Claim 5 is allowable for the same reasons provided above in support of the allowability of Claim 2. Moreover, Smith does not teach or suggest that code characteristics are determined for any specific processor, much less the additional

requirement of "**determining respective program code characteristics <u>for respective</u> <u>ones of a plurality of target processors</u>**", as further required by Claim 5. As such, Claim 5 is not anticipated by Smith. Accordingly, the 35 U.S.C. 102(b) rejection is improper and must be reversed.

Claim 6 further defines the method according to claim 1, wherein said step of deriving comprises selecting one or more pre-defined program code modules in accordance with said program code characteristic from a plurality of available program code modules. Claim 6 is allowable for the same reasons provided above in support of the allowability of Claim 1. Claim 6 is allowable for the same reasons provided above in support of the allowability of Claim 1.

Claim 7 further defines the method according to claim 1, wherein said **program code provides a virtual machine for said target processor.** Claim 7 is allowable for the same reasons provided above in support of the allowability of Claim 1.

Claim 8 further defines the method according to claim 1, wherein said program code comprises elements of a programming language. Claim 8 is allowable for the same reasons provided above in support of the allowability of Claim 1.

Claim 10 further defines the method according to claim 9, wherein said creating program source code for translating between high level code and instructions <u>is for one</u> <u>of a plurality of target processors</u>. Claim 10 is allowable for the same reasons provided above in support of the allowability of Claim 9. In addition to the above, there is no teaching whatsoever in Smith for a "plurality" of target processors. As such, Smith fails to teach or suggest, "creating program source code for translating between high level code and instructions <u>is for one of a plurality of target processors</u>", as required by Claim 10. As such, Claim 10 is not anticipated by Smith. Accordingly, the 35 U.S.C. 102(b) rejection is improper and must be reversed.

Claim 12 further defines the data processing apparatus according to claim 11, further configured for creating program source code for translating between high level code and instructions **for one of a plurality of target processors**. Claim 12 is allowable for the same reasons provided above in support of the allowability of Claim 11. In addition to the above, there is no teaching whatsoever in Smith for a "plurality" of target processors. As such, Smith fails to teach or suggest, "creating program source code for translating between high level code and instructions **for one of a plurality of target processors**", as required by Claim 12. As such, Claim 12 is not anticipated by Smith. Accordingly, the 35 U.S.C. 102(b) rejection is improper and must be reversed.

Claim 14 further defines the apparatus of claim 13, by further comprising at least one additional program code module for translating between high level code and instructions **for respective ones of at least two target processors**. Claim 14 is allowable for the same reasons provided above in support of the allowability of Claim 13. In addition to the above, there is no teaching whatsoever in Smith for a target processor – much less for multiple target processors. As such, Smith fails to teach or suggest, "at least one additional program code module for translating between high level code and instructions **for respective ones of at least two target processors**", as required by Claim 14. As such, Claim 14 is not anticipated by Smith. Accordingly, the 35 U.S.C. 102(b) rejection is improper and must be reversed.

Claim 15 further defines the apparatus according to claim 14, wherein said at least two program code modules are selected from a plurality of predefined program code modules. Claim 15 is allowable for the same reasons provided above in support of the allowability of Claim 14.

Claim 18 further defines the apparatus according to claim 13, wherein said program source code provides a virtual machine for said target processor. Claim 18 is

allowable for the same reasons provided above in support of the allowability of Claim 13.

Claim 19 further defines the apparatus according to claim 14, wherein said program source code provides a virtual machine **for said target processor or processors**. Claim 19 is allowable for the same reasons provided above in support of the allowability of Claim 14. In addition to the above, there is no teaching whatsoever in Smith for a target processor – much less for multiple target processors. As such, Smith fails to teach or suggest, "wherein said program source code provides a virtual machine **for said target processor or processors**", as required by Claim 19. As such, Claim 19 is not anticipated by Smith. Accordingly, the 35 U.S.C. 102(b) rejection is improper and must be reversed.

Claim 20 further defines the apparatus according to claim 13, wherein said program source code comprises elements of a programming language. Claim 20 is allowable for the same reasons provided above in support of the allowability of Claim 13.

Claim 21 further defines the apparatus according to claim 14, wherein said program source code comprises elements of a programming language. Claim 21 is allowable for the same reasons provided above in support of the allowability of Claim 14.

For the above reasons, favorable consideration of the appeal of the Final Rejection in the above-referenced application, and its reversal, are respectfully requested.

Respectfully submitted,

Ronald O. Neerings
Reg. No. 34,227
Attorney for Appellants

TEXAS INSTRUMENTS INCORPORATED
P.O. BOX 655474, M/S 3999
Dallas, Texas 75265
Phone: 972/917-5299
Fax: 972/917-4418

CLAIMS APPENDIX

<u>CLAIMS ON APPEAL</u>:

1.   A method for generating in an apparatus program source code for translating high level code into instructions for a target processor, the method comprising:

determining a program code characteristic corresponding to said target processor;

deriving one or more program code modules in accordance with said program code characteristic; and

generating program source code for translating high level code into instructions for said target processor from said one or more program code modules.

2.   A method according to claim 1, for generating program source code for translating high level code into instructions for one of a plurality of target processors.

3.   A method according to claim 1, comprising forming agglomerated program source code from a plurality of program code modules in accordance with said program code characteristic.

4.   A method according to claim 1, further comprising deriving said program code modules in accordance with a desired functionality for said target processor.

5.   A method according to claim 2, wherein:

said step of determining comprises determining respective program code characteristics for respective ones of a plurality of target processors;

said step of deriving comprises deriving respective program code modules in accordance with said respective program code characteristics; and

said step of generating comprises generating program source code for translating high level code into instructions for said target processors from said program code modules.

6.   A method according to claim 1, wherein said step of deriving comprises selecting one or more pre-defined program code modules in accordance with said program code characteristic from a plurality of available program code modules.

7.   A method according to claim 1, wherein said program code provides a virtual machine for said target processor.

8.   A method according to claim 1, wherein said program code comprises elements of a programming language.

9.   A method for creating in an apparatus program source code for translating between high level code and instructions for a target processor, comprising the steps of:
     determining a program code characteristic corresponding to said target processor;
     selecting one or more predefined program code modules in accordance with said program code characteristic; and
     forming program source code for translating high level code into instructions for said target processor from said selected one or more predefined program code modules.

10.   A method according to claim 9, wherein said creating program source code for translating between high level code and instructions is for one of a plurality of target processors.

11.   A data processing apparatus for creating program source code for translating between high level code and instructions for a target processor, the data processing apparatus being configured to:

determine a program code characteristic corresponding to said target processor identifier input to said data processing apparatus;

derive one or more program code modules in accordance with said program code characteristic; and

create program source code for translating high level code into instructions for said target processor from said derived one or more program code modules.

12.   The data processing apparatus according to claim 11, further configured for creating program source code for translating between high level code and instructions for one of a plurality of target processors.

13.   An apparatus, comprising at least one program source code module of a plurality of program source code modules for translating between high level code and instructions for a target processor, said at least one program code module corresponding to a characteristic of said target processor and being selected from said plurality of program source code modules.

14.   The apparatus of claim 13, further comprising at least one additional program code module for translating between high level code and instructions for respective ones of at least two target processors.

15.   The apparatus according to claim 14, wherein said at least two program code modules are selected from a plurality of predefined program code modules.

18.   The apparatus according to claim 13, wherein said program source code provides a virtual machine for said target processor.

19.   The apparatus according to claim 14, wherein said program source code provides a virtual machine for said target processor or processors.

20.   The apparatus according to claim 13, wherein said program source code comprises elements of a programming language.

21.   The apparatus according to claim 14, wherein said program source code comprises elements of a programming language.

22.   A processor, configured in accordance with program code comprising at least one program source code module of a plurality of program source code modules, for translating between high level code and instructions for a target processor, said at least one program source code module being in accordance with a characteristic of said target processor and selected from said plurality of program source code modules.

23.   A processor, configured by program code comprising an agglomeration of two or more program source code modules of said plurality of said program source code modules.

24.   A system comprising a first and second processor, said first and second processor configured in accordance with program code comprising at least two program source code modules, wherein the first of said at least two program source code modules is arranged to translate high level code to instructions for said first processor and a second of said at least two program source code modules is arranged to translate high level code to instructions for said second processor.

EVIDENCE APPENDIX

No documents are being submitted with the Appeal Brief.

RELATED PROCEEDINGS APPENDIX

Appellants are not aware of pending appeals in related applications.